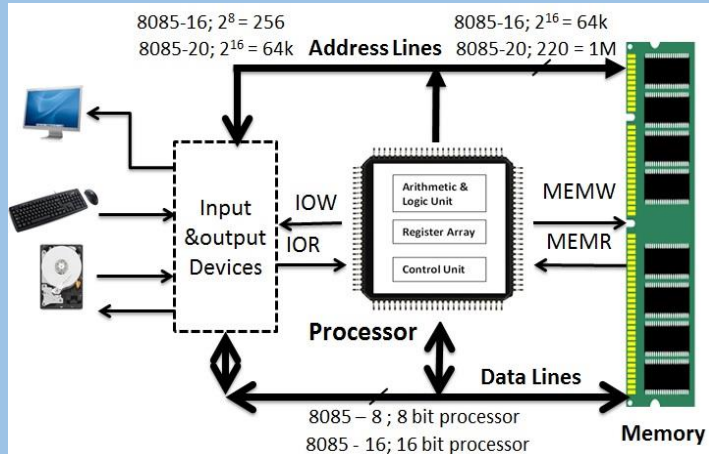


Workshop on Assembly Language Programming



```
section .text
    global main

    main:
        mov     eax, 4      ;system call number (sys_write)
        mov     ebx, 1      ;first argument: file handle (stdout)
        mov     ecx, msg    ;second argument: pointer to message to write
        mov     edx, len    ;third argument: message length
        int     0x80       ;call kernel

        mov     eax, 1      ;system call number (sys_exit)
        mov     ebx, 0      ;first syscall argument: exit code
        int     0x80       ;call kernel

section .data
    msg db      "Hello, world!", 0xa
    len equ    $ - msg
```



By Mr. Kishore Kumar Boddu

India's Leading Embedded
Systems Trainer & Real-time
Embedded Expert

Assembly Language Programming (ALP) Importance

“Embedded knowledge, however good, without knowing assembly language is incomplete. There is always difficulty in solving problems as you don't know how things work.”

**By Kishore Kumar Boddu,
India's Leading Embedded Systems Trainer
& Real-time Embedded Expert.**

10 Reasons to learn Assembly language,

1. Inner working of the computer
2. Understanding the computer architecture and programs for the programmers. CPU Registers, stack trace and Flags to solve the CPU architecture level problems.
3. Operating Systems operations, function call tracing, context switches
4. Communicating with the hardware.
5. How to generate efficient code?
6. How to write Microcontroller programming, bare metal programming & Firmware programming.
7. How computer boot sequence works? What are the roles & responsibilities of boot loader?
8. Reverse engineering techniques like hacking..
9. Which data type to use and when?
10. How to write a device driver?

Learning Assembly Language is the first step to start a career in Embedded Systems.

Computer Architecture

- Micro Computer Structure and its operations
- Types of Computer Architectures
 - Von Neumann vs Harvard architecture
 - RISC vs CISC architecture
 - Little Endian vs Big Endian
 - I/O Mapped I/O vs Memory mapped I/O Architecture
- Embedded Computer vs Desktop Computer

Computer Architecture Programming - Syllabus

8085 Microprocessor

- Microprocessor Specifications
- Memory Mapping
- Microprocessor Signal Description
- Interrupt Vector Table
- Microprocessor Functional Block Diagram
- How Assembly Instruction Works?
- Types of Instruction Cycles
- Addressing modes
- Instruction clock diagram
- Memory Segmentation
- Instruction Pipelining

Intel & ARM Instruction Set

- Data transfer Instructions
- Arithmetical Instructions
- Logical Instructions
- Branch Control Instructions
- Machine Control Instructions

Computer Architecture Programming - Syllabus

Assembly Language Programming

- Assembly Language Fundamentals
- Pseudo Code & Flowchart
- Instructions
- Logical Instructions
- Branch Control Instructions
- Machine Control Instructions

Lab Requirements

- Basic tools used for low-level programming.
- The editor, assembler, linker, loader, debugger, and machine language monitor.
- ALP Simulator compatible with Windows OS

Electronics Graduate Career Goal



Software Engineer



Embedded Software Engineer

Service Based Companies

tcs Infosys® accenture Cognizant

Job Designation
Software Engineer

Salary
Entry Level avg: 3.4 CTC
Exp. Level Avg: No.of Years exp * 2

Job Satisfaction


After 2+ Years of Exp

Join Working Professional training at **Kernel Masters**

Gain 3+ years of Real-time Experience

Semiconductor companies

Qualcomm intel ST BOSCH

Job Designation
Firmware Engineer, System Engineer, R&D Engineer

Salary
Entry Level avg: 3.4 CTC
Exp. Level Avg: No.of Years exp * 4

Job Satisfaction
